Automated Extractive Text Summarization using Genetic and Simulated Annealing Algorithms and their Hybridization

Moheb R. Girgis
Department of Computer Science
Faculty of Science
Minia University
El-Minia, Egypt

Marina Esam
Department of Computer Science
Faculty of Science
Minia University
El-Minia, Egypt

Mamdouh M. Gomaa
Department of Computer Science
Faculty of Science
Minia University
El-Minia, Egypt

ABSTRACT

With the growing world of information, the increase in on-line publishing, and prevalent access to the Internet, huge volume of electronic documents are currently available on-line. Automatic text summarization (ATS) has attracted great interest to assist users and computer systems to process vast amount of texts and extract relevant knowledge in a more efficient way. An ATS system can generate a summary of a document, i.e. short text that includes the main information in it. The aim of this work is to study the performance of ATS systems that utilize metaheuristic and heuristic algorithms in automated extractive text summarization. To this end, this paper proposes Genetic Algorithm (GA)-based, Simulated Annealing (SA)-based, and hybrid GA-SA-based methods for solving the single document summarization (SDS) problem. The objective of these methods is generating a high-quality summary that contains the main information of a given document. In these methods, to assess the quality of solutions (summaries) being generated, an objective function is used that will be maximized. This objective function is represented as a weighted sum that combines five features: sentence position, similarity with title, sentence length, cohesion, and coverage. The paper presents the results of the experiments that have been conducted to evaluate the quality of the summaries generated by the proposed SDS algorithms by applying them to sample articles from the CNN corpus, using co-occurrence statistical metrics (ROUGE metrics) and three content-based metrics (Fitness, Readability and Cohesion).

General Terms

Natural Language Processing, Text Summarization, Metaheuristic Algorithms, Heuristic Algorithms.

Keywords

Single-Document Summarization, Automatic Text Summarization, Extractive Summarization, Genetic Algorithm, Simulated Annealing, ROUGE, Readability, Cohesion, CNN Corpus.

1. INTRODUCTION

With the growing world of information, the increase in on-line publishing, and prevalent access to the Internet, huge volume of electronic documents are currently available on-line. Due to this growth of online information, it has become difficult for users to find relevant information and may get exhausted reading large amount of texts and leave out interesting and important documents. To face such difficulties, automatic text summarization (ATS) has attracted great interest to assist users and computer systems to process vast amount of texts and extract relevant knowledge in a more efficient way. Given a document, an ATS system can generate a summary of this document, i.e. short text that includes the main information in it.

Text summarization is categorized, based on the number of documents, into single- and multi-document summarizations [1]. In a single-document summarization (SDS), information is extracted from a single document, whereas in multi-document summarization (MDS), information is extracted from several documents about the same topic.

In addition, text summarization is categorized, based on the summary results, into extractive and abstractive text summarization [2]. In the extractive summarization, most relevant information is extracted directly from the given text, whereas in the abstractive summarization, the relevant information is rephrased or new sentences are generated from a group of relevant concepts in the given text [3].

The task of generating a summary using ATS can be formulated as an optimization problem. Metaheuristic algorithms, such as Genetic algorithms (GAs), and heuristic algorithms, such as Simulated Annealing (SA), have been successfully applied to solve optimization problems.

So, the aim of this work is to study the performance of ATS systems that utilize metaheuristic and heuristic algorithms in automated extractive text summarization. To achieve that aim, this paper proposes GA-based, SA-based, and hybrid GA-SA-based methods for solving the SDS problem. The objective of these methods is generating a high-quality summary that contains the main information of a given document. In these methods, to assess the quality of solutions (summaries) being generated, an objective function is used that will be maximized. This objective function is represented as a weighted sum that combines five features: sentence position, similarity with title, sentence length, cohesion, and coverage. Experiments have been conducted to evaluate the quality of the summaries generated by the proposed SDS algorithms by applying them to sample articles from the CNN corpus, using co-occurrence statistical metrics (ROUGE metrics with Recall, Precision and F-measure scores) and three content-based metrics (Fitness, Readability and Cohesion).

The paper is organized as follows: Section 2 gives a review of examples of the recent studies related to extractive single-document summarization; Section 3 describes the document representation and the similarity measure; Section 4 describes the summary quality features; Section 5 explains the activities of the document preprocessing stage of the proposed methods; Section 6 describes the proposed GA-based, SA-based, and hybrid GA-SA-based methods for solving the SDS problem; Section 7 presents the results of the experiments that have been conducted to evaluate the quality of the summaries generated by the proposed SDS methods; Finally, Section 8 concludes the presented work.

2. RELATED WORK

Several research studies have been proposed in the field of extractive text summarization. This section gives a review of examples of the recent studies related to extractive single-document summarization.

Nikoo, et al. [4] presented a method for ATS based on the bacterial foraging optimization. Sarkar [5] proposed a method for automatic single document summarization using the key concepts in documents. Mendoza et al. [6] proposed an extractive generic summarization method for single documents by using generic operators and guided local search. This method uses a memetic algorithm which has combined the population-based search of evolutionary algorithm with a guided local search strategy. Mahdipour and Bagheri [7] proposed Persian text summarizer system that employs combination of graph-based and the TF-IDF methods to weight the sentences, and a hybrid algorithm that combines GA and SA for sentence selection to make a summary. The SA-GA algorithm employs SA for crossover operation in GA. The fitness function is based on three factors: Readability Factor, Cohesion Factor, and Topic-Relation. Kikuchi et al. [8] proposed a technique to summarize a single document based on the dependency between words obtained through a dependency parser and dependency between sentences obtained through Rhetorical Structure Theory (RST), where the summarization task has been formulated as an integer linear programming problem. Asgari et al. [9] proposed an approach for text summarization using multi-agent particle swarm optimization. Mirshojaei and Masoomi [10] used cuckoo search optimization algorithm (CSOA) to enhance the performance of extractive-based summarization task. Parveen and Strube [11] proposed a graph-based method for extractive single-document summarization, which considers three properties of summarization: importance, non-redundancy, and local coherence. Hassan [12] proposed a method for automatic single document summarization using Ant Colony Optimization (ACO). Christian et al. [13] proposed a method for a single document summarization using TF-IDF. Sinha et al. [14] proposed a fully data-driven approach for single document summarization using Feed Forward Neural Network (FFNN). Alguliyev et al. [15] proposed a two-stage sentences selection model, called COSUM, for text summarization based on clustering and optimization techniques. At the first stage, to discover all topics in a text, the sentences set is clustered by using k-means method. At the second stage, for selection of salient sentences from clusters, an optimization model that uses an adaptive differential evolution algorithm is employed. Xu and Durrett [16] presented a neural model for single-document summarization based on joint extraction and syntactic compression. Hernández-Castañeda et al. [17] proposed an approach for ATS, which uses semantic features generated by using two methods: Doc2vec and LDA in a GA that searches the best clustering of sentences. This proposed method increases not only the coverage by clustering the sentences to identify the main topics in the source document but also the precision by detecting the keywords in the clusters. El-Kassas et al. [18] proposed an extractive graph-based framework "EdgeSumm" that combines a set of extractive ATS methods: graph-based, statistical-based, semantic-based, and centrality-based methods. Heidary et al. [19] introduced a method for selective text summarization using the GA and generation of repetitive patterns. This method optimizes the vector of the main document's properties in the production of a summary by identifying and extracting the relationship between the main features of the input text and the creation of repetitive patterns. Belwal et al. [20] proposed a graph-based summarization technique, which takes into account the similarity among the individual sentences and the similarity between the sentences and the input document, and incorporates the topic modeling while assigning the weight among the edges of the graph. He et al. [21] presented CTRLSUM, a generic framework to control generated summaries through a set of keywords. During training keywords are extracted automatically without requiring additional human annotations. At test time CTRLSUM features a control function to map control signal to keywords. Anand and Wagh [22] proposed a simple generic technique that uses FFNN for the summarization task for Indian legal judgment documents, which has the advantage of producing an extractive summary without the need to create features or domain knowledge.

This work presents three proposed methods, GA-SDS, SA-SDS, and GASA-SDS, to study the performance of ATS systems that utilize metaheuristic and heuristic algorithms and their hybridization in automated extractive text summarization. The summaries generated by the proposed methods were evaluated using co-occurrence statistical metrics (ROUGE metrics) and three content-based metrics (Fitness, Readability and Cohesion).

3. DOCUMENT REPRESENTATION AND SIMILARITY MEASURE

A document is represented by the set D = {s_1, s_2, …, s_n} where $\mathbf{s_i}$ corresponds to the ith sentence of the document and n is the number of sentences in it. A sentence in the document is represented by the set $s_i = \{t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{im}\}$, where $\mathbf{t_{ik}}$ is the kth term of the sentence $\mathbf{s_i}$ and m is the total number of terms of the whole document. The vector representation of a sentence in the document is $s_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots, w_{im}\}$, where $\mathbf{w_{ik}}$ is the weight of the term $\mathbf{t_k}$ in the sentence $\mathbf{s_i}$. This weight is calculated as the relative frequency of the term in the document and is calculated according to Eq. (1) [23]

$$w_{ik} = \frac{f_{ik}}{MaxFreq_i} \times \log\frac{n}{1+n_k} \quad (1)$$

where $\boldsymbol{f_{ik}}$ is the frequency of the term k in sentence s_i, $\boldsymbol{MaxFreq_i}$ is the number of occurrences of the most frequent term in the sentence $\boldsymbol{s_i}$ and $\boldsymbol{n_k}$ is the number of sentences where the term t_k appears.

The similarity between two sentences s_i and s_j, according to the vector representation described is calculated as the cosine similarity, which is related to the angle of the vectors s_i and S_j, and is calculated according to Eq. (2) [23]

$$sim_{cos}(s_i, s_j) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{(\sum_{k=1}^{m} w_{ik}^2)\,(\sum_{k=1}^{m} w_{jk}^2)}} \quad (2)$$

where $\boldsymbol{m}$ is the total number of terms in the document, $\boldsymbol{w_{ik}}$ refers to the weight of the term k in the sentence s_i, and $\boldsymbol{w_{jk}}$ is the weight of the term k in the sentence s_j.

4. SUMMARY QUALITY FEATURES

The ATS seeks to select the most relevant sentences in a document. A set of features was used, independent of the domain and language, to determine the quality of a summary based on the sentences of which it is comprised. These features are described below.

4.1 Sentence Position

To evaluate a sentence based on its position, a selection criterion is defined that uses the distance that exists between the sentence and the start of the document, assigning greater value to the initial sentences. The position-based metric P of all sentences in summary *S* is calculated by Eq. (3) [24]:

$$P = \sum\nolimits_{\forall s_i \in S} \sqrt{\frac{1}{q_i}} \qquad (3)$$

where ***q**$_i$* indicates the position of the sentence *s$_i$* in the document. In this equation, ***P*** has high values when sentences in summary belong to the first sentences in the document, and ***P*** has low values when sentences in summary belong to the last sentences in the document.

4.2 Similarity of Sentences with Title

This feature is based on the assumption that a good summary contains sentences similar to the title of the document. This similarity is calculated, by Eq. (4), as follows [25]:

$$RT_s = \sum_{\forall s_i \in S} \frac{sim_{cos}(s_i, t)}{nS}$$

$$RTF_s = \frac{RT_s}{\max\limits_{\forall S} RT} \qquad (4)$$

where ***sim$_{cos}$(s$_i$, t)*** is the cosine similarity of sentence *s$_i$* with title t, ***nS*** is the number of sentences in the summary, ***RT$_s$*** is the average of the similarity of the sentences in the summary *S* with the title, ***max RT*** is the average of the maximum values obtained from the similarities of all sentences in the document with the title (i.e. the average top greater nS similarities of all sentences with the title), and ***RTF$_s$*** is the similarity factor of the sentences of the summary *S* with the title. RTF is close to 1 when sentences in summary are closely related to the document title and RTF is close to 0 when sentences in summary are very different to the document title.

4.3 Sentence Length

The length (***L***), normalized by Sigmoid function [26], for the sentences of a summary is calculated, by Eq. (5), as follows:

$$L = \sum\nolimits_{\forall s_i \in S} \frac{1 - e^{-alpha}}{1 + e^{-alpha}}, \; alpha = \frac{l(s_i) - \mu(l)}{std(l)} \qquad (5)$$

where ***l(s$_i$)*** is the length of sentence s$_i$ (measured in words), ***μ(l)*** is the average length of the sentences of the summary, and ***std(l)*** is the standard deviation of the lengths of the sentences of the summary. With this feature, a sentence that is not too short in length will obtain a good grade.

4.4 Cohesion

Cohesion is a feature that determines the degree of relatedness of the sentences that make up a summary. Ideally, the connection between the ideas expressed in the sentences of the summary should be tightly coupled. For its calculation, the cosine similarity measure of one sentence to another is used, see Eq. (6) [25]

$$Coh = \frac{\log(C_s \times 9 + 1)}{\log(M \times 9 + 1)}$$

$$C_s = \frac{\sum_{\forall s_i, s_{j,j>i} \in S} sim_{cos}(s_i, s_j)}{N_s}$$

$$M = maxSim_{cos}(s_i, s_j), \quad i, j \le nS \qquad (6)$$

where ***Coh*** corresponds to the cohesion of a summary, ***C$_s$*** is the average similarity of all sentences in the summary S, ***sim$_{cos}$(s$_i$,s$_j$)*** is the cosine similarity between sentences s$_i$ and s$_j$, *N$_s$* is the number of nonzero similarity relationships in the summary, ***nS*** is the number of sentences in the summary, and ***M*** is the maximum similarity of the sentences in the summary. In this way, Coh tends to zero when the summary sentences are too different among each other, while that Coh tends to one when these sentences are too similar among each other. Thus, this feature tends to favor the summaries that contain sentences about the same topic.

4.5 Coverage

Coverage attempts to measure the extent to which the sentences of a summary provide the reader with the most important information from the original document. Thus, this feature is defined as the similarity between the sentences that make up a summary and the full document, and is calculated, by Eq. (7), as follows [6]:

$$Cov = \sum\nolimits_{\forall s_i \in S} \sum\nolimits_{\forall s_j \in S, j > i} [sim_{cos}(W, s_i) + sim_{cos}(W, s_j)] \qquad (7)$$

where s$_i$ and s$_j$ are the vectors of weights of the terms in the sentences *i* and *j*, respectively, belonging to the summary, *W* = {w$_1$, w$_2$, . . ., w$_k$, . . ., w$_m$}is the vector of weights of the terms in the document, w$_k$ is the weight of the term t$_k$ in the document**,** and m is the total number of terms in the document. The weight w$_k$ is calculated as the relative frequency of the term in the document and is calculated according to Eq. (8)

$$w_k = \frac{f_k}{MaxFreq} \times \log \frac{n}{1 + n_k} \qquad (8)$$

where *f$_k$* is the frequency of the term *t$_k$* in the document, *MaxFreq* is the number of occurrences of the most frequent term in the document, and *n$_k$* is the number of sentences where the term *t$_k$* appears.

5. DOCUMENT PREPROCESSING

Before starting the automatic generation of a summary, a preprocessing of the document is performed, which includes linguistic techniques, such as word tokenization, removal of stop words, upper case and punctuation marks, segmentation of sentences, and stemming [23]. It should be noted that preprocessing is performed on the original document as well as the reference (gold standard) summary.

5.1 Tokenization

In this work, sentences are tokenized into words by using word_tokenize module provided by nltk.tokenize package [27].

5.2 Stopwords Removal

Stopwords are some common words that are present in text but do not contribute in the meaning of a sentence, such as prepositions, articles, pronouns, etc. Such words are not at all important for the purpose of information retrieval or natural language processing (NLP), and are considered noisy terms. So, their removal can be helpful before the execution of any NLP task. Such removal is usually performed by word filtering with the aid of a list of stopwords. This work used the NLTK stopwords corpus [27].

5.3 Segmentation

The segmentation process consists of dividing the text into meaningful units, in this case sentences. In this work, a segmentation tool, developed by the authors, is used.

5.4 Stemming

Stemming is a computational procedure that reduces the words with the same root, or stem, to a common form, eliminating the variable suffixes [Manning et al., 2008]. The Porter stemming algorithm is one of the most common stemming algorithms, which is basically designed to remove and replace well-known suffixes of English words. It is implemented in this work by using PorterStemmer nltk class [27].

5.5 Document Statistics Calculations

The computation of the summary quality features, presented in Sec. 4, requires obtaining the following statistics for the document to be summarized:

- Word-frequency: This includes the number of times a word appears in the document and the number of sentences that word appears in the document.
- Weight vector of each sentence **s_i** {w_{i1}, w_{i2}, . . ., w_{ik}, . . ., w_{im}}, where **w_{ik}** is the weight of the term **t_k** in the sentence. (Eq. 1)
- Document weight vector, W = {w_1, w_2, . . ., w_k, . . ., w_m}, where w_k is the weight of the term t_k in the document (Eq. 8)
- Similarity between all sentences without duplication, i.e., similarity matrix $sim_{cos}(s_i, s_j),\ \ j > i$ (Eq. 2)
- Similarity between the document and each sentence, $sim_{cos}(W, s)$.

6. PROPOSED SINGLE DOCUMENT SUMMARIZATION METHODS

This section describes the proposed GA-based, SA-based, and hybrid GA-SA-based methods for solving the single document summarization (SDS) problem. The objective of each of these methods is generating a high-quality summary of a document that contains the main information of this document.

6.1 Proposed GA-Based Single Document Summarization (GA-SDS) Method

Genetic algorithms (GAs) are powerful search techniques that allow a high-quality solution to be derived from a large search space in polynomial time, by applying the principle of evolution. A GA combines the exploitation of best solutions from past searches with the exploration of new regions of the solution space. Any solution in the search space of the problem is represented by an individual (chromosome). The quality of an individual in the population is determined by a fitness function. The fitness value indicates how good the individual is compared to others in the population.

To use a GA to solve the SDS problem, it is required to determine the representation of individuals in the population, the fitness function, and the genetic operators. The details of the components of the proposed GA-SDS are presented in the following subsections.

6.1.1 GA-SDS problem representation and initial population

In the proposed GA-based single document summarization (GA-SDS) method, a solution (summary) is represented by a chromosome, which is a binary vector. Thus, if a document is composed of n sentences {s1, s2, . . ., sn}, a chromosome is composed of n genes, each representing a sentence in the document, taking the value of 1 if the sentence belongs to the summary represented by the chromosome, or 0 otherwise. For example, if the document consists of 10 sentences (i.e., n = 10), the chromosome [0, 1, 0, 1, 1, 0, 0, 0, 1, 0] represents a summary that is composed of the second, fourth, fifth and ninth sentence of the original document.

Each individual in the initial population is formed by generating a set of random numbers between 1 and n, then a value of 1 is given to the gene (sentence) that corresponds to each random value generated, thereby indicating that this sentence becomes part of the summary represented by the current chromosome. The chromosome representing a summary S must satisfy the condition:

$$\sum_{s_i \in S} l_i \ \leq maxLen \tag{9}$$

where li is the length of the sentence si (measured in words) and maxLen is the maximum number of words allowed in the generated summary.

6.1.2 Fitness function and selection

A fitness function is used to measure the quality of the individuals in the population according to the given optimization objective. In GA-SDS, to assess the quality of a summary represented by a chromosome Sk, an objective function is required, which will be maximized. This objective function is represented as a weighted sum that combines the five features, described by Eqs. (3)-(7), which determine the quality of a summary. That is the objective function is formulated as follows [6]:

$$f(S_k) = \alpha\, P(S_k) + \beta\, RT(S_k) + \gamma\, L(S_k) + \delta\, Coh(S_k) + \rho\, Cov(S_k) \tag{10}$$

where the coefficients α, β, γ, δ, and ρ are the weightings of the 5 features, and satisfy the condition: α + β + γ + δ + ρ = 1.

This objective function is the fitness function that will be used in GA-SDS to evaluate the quality of generated summaries.

6.1.3 Genetic operators

Genetic operations manipulate individuals in the current population and generate new individuals. The proposed algorithm combines the exploitation of the past results by selecting parent chromosomes for reproduction based on their fitness with the exploration of new areas in the search space via crossover and mutation. Chromosomes with better fitness values have a higher probability of contributing one or more offspring in the next generation. The three genetic operators, selection, crossover, and mutation, used in GA-SDS, are described below.

6.1.3.1 Selection operation

Selection is the stage of a GA in which individuals are chosen from a population for later breeding (crossover). In GA-SDS, the roulette wheel selection method [28] is used to select the parents to be mated in the crossover operation.

6.1.3.2 Crossover

During crossover, two parents (chromosomes) exchange chunks of genetic information to produce two new offspring. The objective here is to create a better population over time by combining material from pairs of (fitter) members from the parent population. In GA-SDS, a one-point crossover operator is used. The crossover is applied with a certain crossover rate (X_r), which is the ratio of the number of offspring produced by crossover in each generation to the population size. In a one-point crossover operator, one random cross point is selected and the genes between the cross point to the end of the chromosome is swapped among the two mating chromosomes.

6.1.3.3 Mutation

Mutation operator is used for finding new points in the search space so that population diversity can be maintained. In GA-SDS, a chromosome is selected for mutation based on a mutation probability ***Mr_1*** (chromosome mutation rate). Then, if a chromosome is selected for mutation, it is mutated on a gene-by-gene basis, where a gene is selected at random with a second probability ***Mr_2*** (gene mutation rate) and if it has value 0, it is replaced by 1, otherwise it is not changed. Before mutating (i.e., setting a gene value to 1), the summary length constraint represented by the chromosome is checked based on Eq. (9). If the restriction is not met, the gene is not mutated.

6.1.4 Overall GA-SDS algorithm

The GA-SDS algorithm is given in Figure 1. The input to GA-SDS are a document D = {s_1, s_2, …, s_n}, where n is the number of sentences in it, document statistics, population size pop_size, maximum number of generations Max_Gen, probability of crossover X_r, probabilities of mutation Mr_1 (chromosome mutation rate) and Mr_2 (gene mutation rate), and the features weights α, β, γ, δ, and ρ of the fitness function.

In each generation, the new population is created by repeatedly adding two child chromosomes generated by crossover of parent chromosomes, then mutation, until no new children can be added. If the number of newly added children is less than pop-size, the population is completed by adding chromosomes from the top best parents. Then, the fitness of the new population is evaluated, and the best chromosome is kept.

6.1.5 Decoding (Generation of extractive summary)

After the execution of the GA-SDS algorithm, the chromosome representing the best solution (summary) S_{best} is obtained. The genes of this chromosome with values equal to one represent the sentences of the summary. For example, if gene i is 1, then statement s_i of the original document is part of the summary. These statements are ordered in descending according to the value of the function $f(s_{best,i})$, obtained by Eq. (11) [6], which includes the features of objective function (Eq. (10)). The genes of the chromosome S_{best} are then decoded to obtain the respective sentences of the document, which eventually form the generated summary.

```
GA-SDS Algorithm: GA-based single-document summarization algorithm
Input:    Document D = {$s_1$, $s_2$, …, $s_n$}, where n is the number of sentences in the document
          Document statistics
          pop-size: Population size
          $X_r$: crossover rate
          $Mr_1$ and $Mr_2$: chromosome mutation rate and gene mutation rate, respectively.
          Max_Gen: Maximum number of generations.
          α, β, γ, δ, and ρ: the fitness function features weights.
Output: best summary
Begin
Step1.  Randomly, generate pop-size different chromosomes such that each chromosome
        represents a legal summary, i.e. satisfy length condition (Eq. 9).
Step2.  Set current population = initial population.
Step3.  Set gen =1
Step4.  While gen ≤ Max_Gen Do
          4.1  Evaluate the fitness of the current population using Eq. 10, and keep the
               best chromosome according to the fitness value.
          4.2  Select parent chromosomes using the roulette wheel selection method.
          4.3  Perform crossover and mutation operations to obtain new population.
               (Each generated chromosome is checked, and if it does not represent a
               legal summary, i.e. satisfy length condition (Eq. 9)), it will be rejected.
          4.4  Set current population = new population.
          4.5  Set gen = gen+1.
        End while
Step5.  Return the best chromosome (best summary).
```

Figure 1: The proposed GA-SDS algorithm

$$f(s_{best,i}) = \sqrt{\frac{1}{q_i}} + sim_{cos}(s_i,t) + \frac{1- e^{-alpha}}{1+ e^{-alpha}} + \sum_{s_j \in S_{best}} sim_{cos}(s_i,s_j) + \sum_{j=i+1}^{n} [sim_{cos}(W,s_i) + sim_{cos}(W,s_j)] \tag{11}$$

where $S_{best,i}$ is the ith sentence of the document represented by the ith gene in the chromosome S_{best} whose value is 1, *W* is the document weight vector, t is the title, and $alpha = \frac{l(s_i) - \mu(l)}{std(l)}$, as in Eq. 5.

6.2 Proposed SA-based single document summarization (SA-SDS) method

Simulated Annealing (SA) developed by Metropolis et al. [29] is a powerful optimization algorithm. The basic idea behind SA is to start with an initial solution, and iteratively improve it by making small changes to it. At each iteration, the algorithm evaluates the new solution and decides whether to accept or reject it based on a probability function. The probability function is designed to allow the algorithm to escape local optima and explore the solution space.

Solving the SDS problem by using SA requires the determination of the solution (summary) representation, the annealing schedule, the neighborhood operator to generate a new solution from the current solution, and a suitable objective function. The proposed SA-SDS's components are presented below.

6.2.1 Solution representation

The solution representation used in the proposed SA-SDS is the chromosome representation used by GA-SDS.

6.2.2 Initial temperature and annealing schedule

The initial temperature for the search is provided as a parameter and gradually decreases with the progress of the search. The annealing schedule is used to control the probability of accepting a worse solution, as it is implemented as a function of the current temperature.

6.2.3 Neighborhood operator

This operator is used to generate a new solution by making small changes to the current solution. In the proposed SA-SDS, the mutation operator of GA-SDS is used as a neighborhood operator.

6.2.4. Fitness function

The fitness function used to assess the quality of the solution generated by SA-SDS is the same objective function, used by GA-SDS (Eq. 10).

6.2.5. Overall SA-SDS algorithm

The SA-SDS algorithm is given in Figure 2. In steps 1-2, SA-SDS generates an initial solution (chromosome), S_c, and evaluates its fitness using Eq. (10). Next, steps 3-18 includes the main steps of the SA-SDS algorithm. In step 5, the current temperature *Temper* is set to the initial temperature *Init_Temper*. Steps 6-17 includes the outer loop of the SA-SDS algorithm, which repeatedly decreases the temperature by the cooling rate CR, until the stopping criterion is reached. Here, the stopping criterion is *Temper* = Final_Temper, where Final_Temper is calculated as follows:

Final_Temper = Init_Temper /

(1 + Max_N * Init_Temper * CR)

For each temperature, an inner loop (Steps 7-15) is executed *Max_N* iterations. In each iteration, a neighboring valid solution S_n is generated by applying the mutation operator. S_n is accepted as the new current solution, if the difference $\Delta f =$ $f(S_n) - f(S_c)$ is greater than zero, i.e. the new solution is better. If $\Delta f \leq 0$, i.e. the new solution is worse, then accept it with a probability, which is a function of Tempr, $e^{-\Delta f/Temper}$. This probabilistic acceptance is achieved by generating a random number in [0, 1), and if it is less than $e^{-\Delta f/Temper}$, then replace the current solution by the new one. Finally, the best chromosome (best summary) is returned in step 19.

SA-SDS Algorithm: SA-based single-document summarization algorithm

```
Input:    Document D = {s_1, s_2, …, s_n}, where n is the
          number of sentences in the document
          Document statistics
          M_r: mutation rate.
          Maximum no. of iterations Max_N;
          Initial temperature Init_Temper > 0;
          The cooling rate CR;
          $\alpha, \beta, \gamma, \delta$ and $\rho$: the fitness function
          features weights.
Output:   best summary
Begin
  1.  Generate an initial solution S_c (chromosome) at
      random;
  2.  Evaluate S_c (calculate f(S_c) using Eq. (10))
  3.  Apply SA(S_c)
  4.  Begin
  5.    Temper= Init_Temper
  6.    Repeat
  7.      For n = 1 To Max_N Do
  8.        Generate a new valid solution S_n, a
            random neighbor of S_c, using mutation
            operator;
  9.        Calculate f(S_n), using Eq. (10);
            // Compare the change in objective
            // function
  10.       Set $\Delta f =$ f(S_n) - f(S_c)
            // if the new solution is better, accept it
  11.       If $\Delta f \leq 0$ Then
  12.         S_c ← S_n   // S_n replaces S_c
            // if the new solution is worse, accept it
            // with a probability
  13.       Else if random(0,1) < $e^{-\Delta F/Temper}$ Then
  14.         S_c ← S_n
  15.     End For
          // decrement the temperature
  16.     Temper = Temper × CR;
  17.   Until stopping criterion is true;
  18. End
  19. Return the Best Summary;
End
```

Figure 2: The proposed SA-SDS algorithm

6.3 Proposed hybrid GA-SA-based single document summarization (GASA-SDS) method

GA and SA both independently are valid approaches toward problem solving. However, they both have their own strengths and weaknesses. While GA can begin with a population of solutions in parallel, it has poor convergence properties. SA, on the other hand, has better convergence properties but cannot exploit parallelism. The hybrid GASA-SDS blends both these approaches into a single approach in order to retain the strengths of both.

The code of the GASA-SDS Algorithm is the same as the GA-SDS Algorithm, shown in Figure 1, except that, at step 4.3, after performing crossover and mutation operations on the current population, the SA algorithm is applied to the individuals of the new population based on a probability SAr in an attempt to improve the solutions.

7. EXPERIMENTS

This section presents the results of the experiments that have been conducted to evaluate the quality of the summaries generated by the proposed GA-SDS, GAGLS-SDS, SA-SDS, and GASA-SDS methods.

7.1 Dataset

In the experiments, the CNN corpus was used. The CNN corpus [30] is a collection of news document for single-document summarization based on the news articles from the CNN website (http://www.cnn.com). The utilized version of this corpus consists of 2000 articles in English distributed into twelve subject categories, originally tagged by CNN: Business, Health, Justice, Living, Opinion, Politics, Showbiz, Sports, Technology, Travel, United Stated, and world news. One important aspect of this corpus is the presence of a good-quality abstractive summary for each document written by the original authors, called highlights, and an extractive summary, called gold standard, developed by a team of experts based on the highlights, using a computer-assisted methodology. The experiments reported here were performed using the gold-standards of the CNN corpus. Five articles were selected from five categories in the CNN corpus: Justice, business, Health, Sports, and. Technology.

As the articles in CNN corpus are written in XML, an XML parser is used to extract the article title and sentences, and sentences in reference summary (gold standard), of each article, before submitting it to the preprocessing step.

7.2 Summary quality metrics

To evaluate the summaries generated by the proposed SDS methods, referred to as candidates, two types of metrics were used: the co-occurrence statistical metrics and content-based metrics.

7.2.1. Co-occurrence statistical metrics

N-gram co-occurrence statistical measure of ROUGE toolkit was used [31] to evaluate the candidate summary against the reference summary. If p is the number of common N-grams between candidate and reference summary, and q is the number of N-grams extracted from the reference summary only, the ROUGE-N score is computed as follows:

$$ROUGE - N = \frac{p}{q} \qquad (12)$$

where N is the size of the N-gram which can be unigram, bi-gram or trigram. In addition to ROUGE-N, the ROUGE-L score is used, which measures the Longest Common Subsequence (LCS) words between reference and candidate summaries. LCS refers to word tokens that are in sequence, but not necessarily consecutive. The updated ROUGE evaluation methods can generate three types of scores for a candidate summary: Recall, Precision, and F-measure. The recall score is the ratio of common unigrams in the candidate summary and reference summary to the total unigrams in the reference summary. The precision score is the ratio of common unigrams in the candidate summary and reference summary to the total unigrams in the system summary. F- measure indicates how much contents are accurately extracted with respect to both the candidate summary and reference summary by giving them equal weights.

7.2.2. Content-based evaluation metrics

The content-based evaluation metrics used are fitness, cohesion and readability metrics. The candidate fitness is evaluated using Eq. 10, and the cohesion metric is evaluated using Eq. 6. Readability metric [25] can be calculated by obtaining the relatedness between each two adjacent sentences in the summary, using the cosine similarity as follows:

$$\text{Readability}(S) = \frac{\sum_{i=1}^{nS-1} Sim_{cos}(s_i\ ,s_{i+1})}{maxSim_{cos}(s_i,s_{i+1})} \qquad (13)$$

where $\boldsymbol{nS}$ is the number of sentences in the summary S, and $maxSim_{cos}(s_i, s_{i+1})$ is the maximum similarity of the successive sentences in the summary.

7.3 Experimental Results

This section presents the results of applying the proposed GA-SDS, SASDS and GASA-SDS methods to the selected CNN corpus articles. Each method was applied 10 times to each article, then the metrics results were averaged. The maximum length of the generated summaries was 100 words. The proposed methods were implemented on Intel(R) Core(TM) i5-3230M CPU @ 2.60 GHz, with 4 GB of RAM, and Windows 8, 64-bit OS.

The weights used for the objective function (Eq. 10) were: α = 0.35, β = 0.35, γ = 0.29, δ = 0.005, ρ = 0.005, as in [6]. GA-SDS parameters were Max_Gen = 20, pop-size = 10, crossover rate X_r = 0.9, chromosome mutation rate Mr_1 = 0.4, and gene mutation rate Mr_2 = 1.0 / n_bits, where n_bits is the chromosome length. SA parameters were: maximum no. of iterations per temperature Max_N = 10, Initial temperature Init_Temper = 0.9, the cooling rate CR = 0.5, and the probability of applying SA in GASA-SDS SA_r = 1.

7.3.1. Co-occurrence-based metrics results

The ROUGE-1 scores for the proposed SDS methods are shown in Table 1. As shown in the table, the highest Recall, Precision and F-measure scores, for the 1st and 4th articles, were obtained by GASA-SDS; for the 2nd and 3rd articles, were obtained by SA-SDS; and for the 5th article, were obtained by GA-SDS. Figure 3 shows the graphical representation of average ROUGE-1 Recall, Precision and F-measure scores. This figure shows that GA-SDS obtained the highest average ROUGE-1 Recall, Precision and F-measure scores.

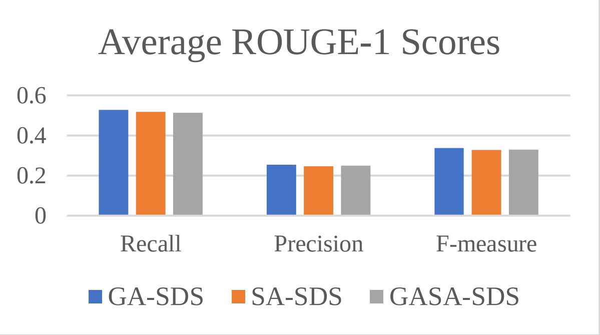

Figure 3. Average ROUGE-1 Recall, Precision and F-measure scores

Table 1. ROUGE-1 scores for the proposed SDS methods

	GA-SDS			SA-SDS			GASA-SDS		
Article	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
ArneDuncan	0.474	0.351	0.403	0.478	0.347	0.402	**0.485**	**0.353**	**0.409**
bitcoinChina	0.503	0.201	0.287	**0.510**	**0.204**	**0.292**	**0.510**	0.203	0.290
softDrinks	**0.538**	0.195	0.287	**0.538**	**0.196**	**0.287**	0.508	0.185	0.271
ChelseaTitleHopes	0.511	0.297	0.375	0.454	0.261	0.332	**0.535**	**0.307**	**0.390**
DesignerGenes	**0.614**	**0.228**	**0.332**	0.608	0.225	0.328	0.530	0.196	0.286
Average	**0.528**	**0.254**	**0.337**	0.518	0.247	0.328	0.514	0.249	0.329

Table 2. ROUGE-2 scores for the proposed SDS methods

	GA-SDS			SA-SDS			GASA-SDS		
Article	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
ArneDuncan	**0.376**	**0.278**	**0.319**	0.206	0.149	0.172	0.240	0.174	0.202
bitcoinChina	0.390	0.153	0.220	0.382	0.151	0.216	**0.392**	**0.154**	**0.221**
softDrinks	0.331	0.118	0.174	**0.333**	**0.120**	**0.176**	0.317	0.113	0.167
ChelseaTitleHopes	0.352	0.203	0.257	0.263	0.149	0.190	**0.366**	**0.209**	**0.266**
DesignerGenes	**0.353**	**0.129**	**0.189**	0.347	0.126	0.185	0.192	0.070	0.102
Average	**0.360**	**0.176**	**0.232**	0.306	0.139	0.188	0.301	0.144	0.192

Table 3. ROUGE-l scores for the proposed SDS methods.

	GA-SDS			SA-SDS			GASA-SDS		
Article	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
ArneDuncan	**0.399**	**0.295**	**0.339**	0.282	0.205	0.237	0.334	0.244	0.282
bitcoinChina	**0.448**	**0.179**	**0.256**	0.443	0.177	0.253	0.430	0.171	0.245
softDrinks	**0.454**	**0.165**	**0.242**	0.424	0.155	0.227	0.422	0.153	0.225
ChelseaTitleHopes	0.432	0.251	0.317	0.347	0.199	0.253	**0.446**	**0.256**	**0.325**
DesignerGenes	**0.443**	**0.165**	**0.240**	**0.443**	0.164	0.239	0.316	0.117	0.170
Average	**0.435**	**0.211**	**0.279**	0.388	0.180	0.242	0.390	0.188	0.249

The ROUGE-2 scores for the proposed SDS methods are shown in Table 2. As shown in the table, the highest Recall, Precision and F-measure scores, for the 1st and 5th articles, were obtained by GA-SDS; for the 2nd and 4th articles, were obtained by GASA-SDS; and for the 3rd article, were obtained by SA-SDS. Figure 4 shows the graphical representation of average ROUGE-2 Recall, Precision and F-measure scores. This figure shows that GA-SDS obtained the highest average ROUGE-2 Recall, Precision and F-measure scores.

The ROUGE-l scores for the proposed SDS methods are shown in Table 3. As shown in the table, the highest Recall, Precision and F-measure scores, for the 3rd article, were obtained by GASA-SDS; for the other 4 articles, were obtained by GA-SDS; and SA-SDS obtained the same highest Recall, for the 5th article, as GA-SDS.

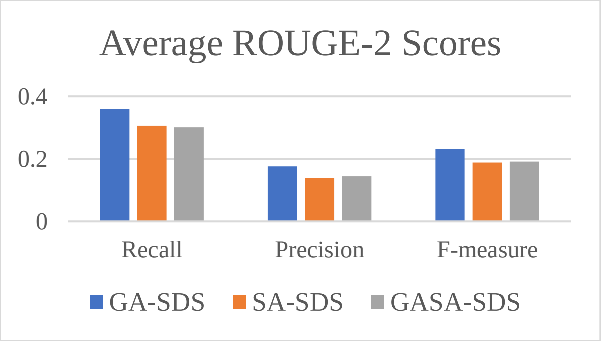

Figure 4. Average ROUGE-2 Recall, Precision and F-measure scores

Figure 5 shows the graphical representation of average ROUGE-l Recall, Precision and F-measure scores. This figure shows that GA-SDS obtained the highest average ROUGE-l Recall, Precision and F-measure scores.

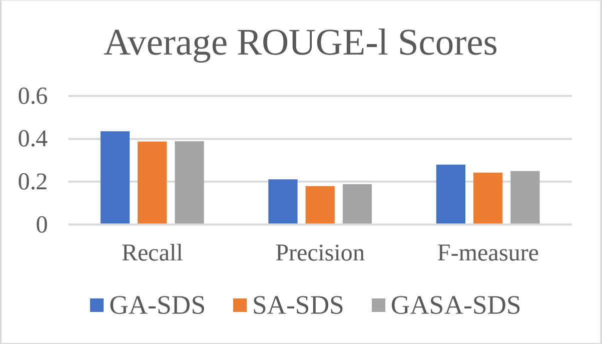

Figure 5. Average ROUGE-l Recall, Precision and F-measure scores

The above co-occurrence-based metrics results show that, on average, GA-SDS produced better summaries than SA-SDS and GASA-SDS, as indicated by all ROUGE scores.

7.3.2. Content-based metrics results

The content-based metrics results for the proposed SDS methods are shown in Table 4. As shown in the table, the highest Fitness, Readability and Cohesion scores, for the 1st article, were obtained by GA-SDS; for the 2nd and 4th articles, the highest Fitness were obtained by GASA-SDS, the highest Readability were obtained by GA-SDS, and the highest Cohesion were obtained by SA-SDS; and for the 5th article, the highest Fitness were obtained by GASA-SDS, and the highest Readability and Cohesion were obtained by SA-SDS. Figure 6 shows the graphical representation of the average Fitness, Readability and Cohesion scores. This figure shows that GA-SDS obtained the highest average Readability score, SA-SDS obtained the highest average Cohesion score, and GASA-SDS obtained the highest average Fitness score.

8. CONCLUSION

This paper proposed GA-based, SA-based, and hybrid GA-SA-based methods for automated extractive text summarization of single-documents. To assess the quality of summaries being generated by these methods, an objective function was used, which is a weighted sum that combines five features: sentence position, similarity with title, sentence length, cohesion, and coverage.

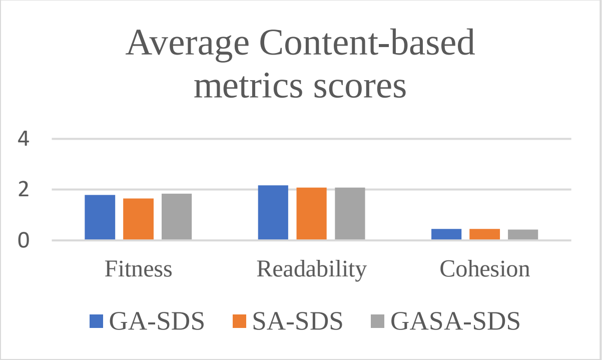

Figure 6. Average Fitness, Readability and Cohesion scores

Experiments have been conducted to evaluate the quality of the summaries generated by the proposed SDS algorithms by applying them to sample articles from the CNN corpus, using co-occurrence statistical metrics (ROUGE metrics with Recall, Precision and F-measure scores) and three content-based metrics (Fitness, Readability and Cohesion). The co-occurrence-based metrics experimental results showed that, on average, GA-SDS produced better summaries than SA-SDS and GASA-SDS, as indicated by all ROUGE scores. On the other hand, the content-based metrics experimental results showed that, on average, GA-SDS produced summaries with better Readability, SA-SDS produced summaries with better Cohesion, and GASA-SDS produced summaries with better Fitness.

In the future work, we intend to study the use of the proposed text summarization methods to improve the performance (in terms of memory and time) of data mining algorithms on IoT data.

Table 4. Content-based metrics results for the proposed SDS methods

	GA-SDS			SA-SDS			GASA-SDS		
Article	**Fitness**	**Readability**	**Cohesion**	**Fitness**	**Readability**	**Cohesion**	**Fitness**	**Readability**	**Cohesion**
ArneDuncan	**1.517**	**2.498**	**0.396**	1.383	2.171	0.376	1.513	2.487	0.375
bitcoinChina	2.212	**1.762**	0.272	2.080	1.699	**0.289**	**2.287**	1.640	0.267
softDrinks	1.672	2.057	0.525	1.567	2.111	0.521	**1.737**	**2.413**	**0.545**
ChelseaTitleHopes	1.465	**2.136**	0.529	1.313	1.934	**0.546**	**1.526**	1.707	0.526
DesignerGenes	2.097	2.375	0.513	1.900	**2.461**	**0.549**	**2.114**	2.189	0.445
Average	1.792	**2.165**	0.447	1.649	2.075	**0.456**	**1.835**	2.087	0.432

9. REFERENCES

[1] Gambhir, M. and Gupta, V. 2017 Recent automatic text summarization techniques: a survey. Artificial Intelligence Review 47, 1–66.

[2] Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur ,A., Affandy, A., Setiadi, D. I. M. 2022 Review of automatic text summarization techniques & methods. Journal of King Saud University – Computer and Information Sciences 34, 1029–1046.

[3] Verma, P., Om, H. 2019 MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. Expert Systems with Applications 120 (5 April 2019), 43-56.

[4] Nikoo, M. D., Faraahi, A., Hashemi, S. M., and Erfani, S. H. 2012 A method for text summarization by bacterial foraging optimisation algorithm. IJCSI Int. J. Comput. Sci. 9 (4), 36–40.

[5] Sarkar, K. 2013 Automatic single document text summarization using key concepts in documents. Journal of Information Processing Systems (JIPS) 9, 602-620.

[6] Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., and Leon, E. 2014 Extractive single-document summarization based on genetic operators and guided local search. Expert Systems with Applications 41, 4158–4169.

[7] Mahdipour, E. and Bagheri, M. 2014 Automatic Persian Text Summarizer Using Simulated Annealing and Genetic Algorithm. International Journal of Intelligent Information Systems. Special Issue: Research and Practices in Information Systems and Technologies in Developing Countries 3 (6-1), 84-90.

[8] Kikuchi, Y., Hirao, T., Takamura, H., Okumura, M., and Nagata, M. 2014 Single document summarization based on nested tree structure. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics 2 (June 23-25), 315–320, Baltimore, Maryland, USA.

[9] Asgari, H., Masoumi, B., and Sheijani, O. S. 2014 Automatic text summarization based on multi-agent particle swarm optimization. In Proceedings of Iranian Conference on Intelligent Systems (ICIS). IEEE, 1–5.

[10] Mirshojaei, S. H. and Masoomi, B. 2015 Text summarization using cuckoo search optimization algorithm. J. Comput. Robot. 8 (2), 19–24.

[11] Parveen, D. and Strube, M. 2015 Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015). AAAI Press, 1298–1304.

[12] Hassan, O. F. 2015 Text Summarization using Ant Colony Optimization Algorithm. Doctoral dissertation. Sudan University of Science and Technology.

[13] Christian, H., Agus, M. P., and Suhartono, D. 2016 Summarization using term frequency inverse document frequency (TF-IDF). ComTech, 7 (4), 285-294.

[14] Sinha, A., Yadav, A., and Gahlot, A. 2018 Extractive Text Summarization using Neural Networks. arXiv:1802.10137 [cs.CL].

[15] Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., and Idris, N. 2019 COSUM: Text summarization based on clustering and optimization. Expert Systems, 36(1), 1–17.

[16] Xu, J. and Durrett, G. 2019 Neural Extractive Text Summarization with Syntactic Compression. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 3292–3303, Hong Kong, China. Association for Computational Linguistics.

[17] Hernández-Castañeda, Á., García-Hernández, R. A., Ledeneva, Y. and Millán-Hernández, C. E. 2020 Extractive Automatic Text Summarization Based on Lexical-Semantic Keywords, IEEE ACCESS 8, 49896-49907.

[18] El-Kassas, W. S., Salama, C. R., Rafea, A. A., and Mohamed, H. K. 2020 EdgeSumm: Graph-based framework for automatic text summarization. Information Processing and Management 57(6), 102264.

[19] Heidary, E., Parvïn, H., Nejatian, S., Bagherifard, K., Rezaie, V., Mansor, Z., and Pho, K. 2021 Automatic Text Summarization Using Genetic Algorithm and Repetitive Patterns, Computers, Materials & Continua (CMC) 67 (1), 1085-1101.

[20] Belwal, R. C., Rai, S., and Gupta, A. 2021 A new graph-based extractive text summarization using keywords or topic modeling. Journal of Ambient Intelligence and Humanized Computing 12(10), 8975–8990.

[21] He, J., Kryscinski, W., McCann, B., Rajani, N., and Xiong, C. 2022 CTRLsum: Towards generic controllable text summarization. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (December 2022), Abu Dhabi, United Arab Emirates, 5879—5915.

[22] Anand, D. and Wagh, R. 2022 Effective deep learning approaches for summarization of legal texts. Journal of King Saud University - Computer and Information Sciences 34 (5), 2141-2150.

[23] Manning, C. D., Raghavan, P., and Schtze, H. 2008 Introduction to information retrieval. Cambridge University Press.

[24] Bossard, A., Genereux, M., and Poibeau, T. 2008 Description of the LIPN systems at TAC 2008: Summarizing information and opinions. In Proceedings of Notebook papers and results, text analysis conference (TAC-2008).

[25] Shareghi, E. and Hassanabadi, L. S. 2008 Text summarization with harmony search algorithm-based sentence extraction. In Proceedings of the 5th international conference on soft computing as transdisciplinary science and technology, 226–231. Cergy-Pontoise, France: ACM.

[26] Gupta, V., Chauhan, P., and Garg, S. 2012 An Statistical Tool for Multi-Document Summarization. International Journal of Scientific and Research Publications 2 (5).

[27] NLTK, https://www.tutorialspoint.com/natural_language_toolkit/index.htm, 20/12/2021

[28] Goldberg, D. E. 1989 Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley.

[29] Metropolis, N., Rosenbluth, Rosenbluth, A. M., Teller, A., Teller, E. 1953 Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics 21 (6), 1087–1092.

[30] Lins, R. D., Simske, S. J., de Souza Cabral, L., de França Silva, G., Lima, R., Mello, R. F., and Favaro, L. 2012 A multi-tool scheme for summarizing textual documents. In Proceedings of the 11st IADIS international conference www/internet 2012, 409–414.

[31] Lin C. 2004 ROUGE: A package for automatic evaluation of summaries. Annual Meeting of the Association for Computational Linguistics (25 July 2004).